UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/824,992	04/02/2001	Kevin J. McGrath	5500-66300	1338

7590 02/23/2004

Lawrence J. Merkel
Conley, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	5

DATE MAILED: 02/23/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/824,992	MCGRATH, KEVIN J.	
	Examiner	Art Unit	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 July 2001 and 23 July 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *02 April 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 2.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-26 have been examined.

Papers Received

2. Receipt is acknowledged of Information Disclosure Statement, request for corrected receipt, and preliminary amendment papers submitted, where the papers have been placed of record in the file.

Oath/Declaration

3. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

> The oath or declaration is defective because:
> It states that priority to a provisional application is not applicable when there is in fact a claim for priority to provisional application 60/224,368.

Drawings

4. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the configuration register must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 2-5, 9-12, 16-19, and 22-25 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7. In claims 2, 4, 9, 16, 18, and 22 the examiner does not know what the "configuration register" is and there is no indication of it in the specification. The examiner will take the broadest reasonable interpretation of the register to be a register that holds any sort of configuration or that may control something.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-3, 5-10, 12-17, 19-23, and 25-26 are rejected under 35 U.S.C. 102(b) as being anticipated by Turley (Advanced 80386 Programming Techniques).

10. In regard to claim 1, Turley discloses a processor comprising:

a. A circuit configured to generate an indication of a default operand size; Page 45 is the beginning of a chapter on memory segmentation of an Intel 80386 processor circuit. Within this chapter on page 52 it is shown that an indication of default operand size is given in a segment descriptor by a "D bit".

b. And an execution core coupled to receive a first instruction, wherein the execution core is configured to override the default operand size with a second

operand size responsive to the first instruction having an implicit stack pointer reference. It is inherent that the Intel 80386 architecture has an execution core to execute the instruction set that page 10 defines. As shown above, page 52 shows that a D bit gives a default operand size. This section, which starts on page 48, shows that this D bit is in each segment descriptor and therefore specifies the default operand size for each segment descriptor. On pages 266-267, the section entitled "Stack Sizing," shows that in the case of an implicit stack operation, the D bit of a *stack* segment descriptor, specifies the address size (and default operand size as shown immediately above for each segment descriptor) instead of the D bit of a *code* segment descriptor as is done for all other instructions. The examiner is taking an instruction with such an implicit stack pointer reference to be the first instruction. This means that the D bit of the stack segment descriptor overrides the default operand size given by the code segment descriptor responsive to the first instruction having implicit stack pointer references.

11. In regard to claim 2, Turley discloses the processor as recited in claim 1 further comprising a segment register and a configuration register coupled to the circuit, wherein the segment register is configured to store a segment selector locating a segment descriptor which includes a first operating mode indication and a second operating mode indication, and wherein the configuration register is configured to store an indication, and wherein the circuit is configured to generate the indication of the default operand size responsive to the first operating mode indication, the second

operating mode indication, and the indication in the configuration register. Page 63 in the last paragraph shows that a segment register holds a selector that identifies a segment descriptor in a descriptor table. As shown above, page 52 shows that the segment descriptor has a D bit, which indicates the default operand size to use (16 or 32 bit), and thus specifies (and is responsive to) first and second operating modes specifying whether to operate in 16 or 32 bit default operand size mode. Page 26 shows that there is a Control (configuration) Register 0. It is further shown that this register contains a PE bit that specifies Real or Protected Mode for processor operation. Pages 8-9 show that segmentation is used in Protected Mode and that the reference almost exclusively uses Protected Mode, which covers the segmentation and overriding principles already discussed, until chapter 9. Page 407 in the last paragraph shows that Real Mode was not designed to use these segmentations features of Protected Mode. Therefore, since the Protected Mode is required for the above functionality to work, the indication in the stated control (configuration) register that specifies this mode controls the default operand size, which then is responsive to the configuration register. Also, the first paragraph of page 408 shows that one needs to override the default 16-bit operations (and operands) to use 32 bits ones. Therefore the default operand size is also responsive to the configuration register because the default for Real mode is always 16 bit operands.

12. In regard to claim 3, Turley discloses the processor as recited in claim 2 wherein the default operand size is 32 bits. Turley has shown on page 52 that the default operand size is 32 bits when the D bit is 1.

13. In regard to claim 5, Turley discloses the processor as recited in claim 3 wherein the second operand size is 64 bits. Page 15 shows that figure 1-3 on page 16 gives the data types for use in the processor. It goes on to say that three most common types of operands are the top three. This means that though less common, the other data types are also operands. With this knowledge at hand, figure 1-3 shows that a long integer is a 64-bit operand. For an instruction that uses this data type, this is the second operand size since it is not the default operand size of 32 bits and thus would need to be overridden as specified above.

14. In regard to claim 6, Turley discloses the processor as recited in claim 1 wherein the execution core is configured to override the default operand size with the second operand size in an absence of an operand size override encoding for the first instruction. As shown above, the D bit of the stack segment register specifies the operand size to use for implicit stack operations. This value is not an instruction encoding but actually an overriding default size for implicit operand purposes.

15. In regard to claim 7, Turley discloses the processor as recited in claim 1 wherein the execution core is coupled to receive a near branch instruction, and wherein the execution core is configured to override the default operand size for the near branch instruction with the second operand size. Page 54 shows that CALL instructions are implicit stack operations. It is inherent in the x86 architecture that CALL instructions are comprised of near and far ("normal" and far, page 180) CALLs and that calls branch to a different piece of code, where near calls branch to code within the current segment and far calls branch to other segments ("near" and "far" pointers, page 15). Because all

CALLs are implicit and CALLs include branches, the stack segment operand size overrides the code segment default operand size for near branches.

16. In regard to claim 8, Turley discloses a method comprising:

a. Generating a default operand size for instructions; Page 45 is the beginning of a chapter on memory segmentation of an Intel 80386 processor circuit. Within this chapter on page 52 it is shown that an indication of default operand size for instructions is given in a segment descriptor by a "D bit".

b. Overriding the default operand size with a second operand size for a first instruction responsive to the first instruction having an implicit stack pointer reference. It is inherent that the Intel 80386 architecture has an execution core to execute the instruction set that page 10 defines. As shown above, page 52 shows that a D bit gives a default operand size. This section, which starts on page 48, shows that this D bit is in each segment descriptor and therefore specifies the default operand size for each segment descriptor. On pages 266-267, the section entitled "Stack Sizing," shows that in the case of an implicit stack operation, the D bit of a *stack* segment descriptor, specifies the address size (and default operand size as shown immediately above for each segment descriptor) instead of the D bit of a *code* segment descriptor as is done for all other instructions. The examiner is taking an instruction with such an implicit stack pointer reference to be the first instruction. This means that the D bit of the stack segment descriptor overrides the default operand size given by the code

segment descriptor responsive to the first instruction having implicit stack pointer references.

17. In regard to claim 9, Turley discloses the method as recited in claim 8 further comprising generating an operating mode responsive to a first operating mode indication in a segment descriptor and a second operating mode indication in the segment descriptor and further responsive to an indication in a configuration register, wherein the generating the default operand size is responsive to generating the operating mode. Page 63 in the last paragraph shows that a segment register holds a selector that identifies a segment descriptor in a descriptor table. As shown above, page 52 shows that the segment descriptor has a D bit, which indicates the default operand size to use (16 or 32 bit), and thus specifies (and is responsive to) first and second operating modes specifying whether to operate in (generate) 16 or 32 bit default operand size mode. Therefore, the default operand size is generated responsive to the generation of the operating mode, which specifies the default operand size. Page 26 shows that there is a Control (configuration) Register 0. It is further shown that this register contains a PE bit that specifies Real or Protected Mode for processor operation. Pages 8-9 show that segmentation is used in Protected Mode and that the reference almost exclusively uses Protected Mode, which covers the segmentation and overriding principles already discussed, until chapter 9. Page 407 in the last paragraph shows that Real Mode was not designed to use these segmentations features of Protected Mode. Therefore, since the Protected Mode is required for the above functionality to work, the indication in the stated control (configuration) register that specifies this mode controls

the default operand size, which then is responsive to the configuration register. Also, the first paragraph of page 408 shows that one needs to override the default 16-bit operations (and operands) to use 32 bits ones. Therefore the default operand size is also responsive to the configuration register because the default for Real mode is always 16 bit operands.

18. In regard to claim 10, Turley discloses the method as recited in claim 9 wherein the default operand size is 32 bits. Turley has shown on page 52 that the default operand size is 32 bits when the D bit is 1.

19. In regard to claim 12, Turley discloses the method as recited in claim 10 wherein the second operand size is 64 bits. Page 15 shows that figure 1-3 on page 16 gives the data types for use in the processor. It goes on to say that three most common types of operands are the top three. This means that though less common, the other data types are also operands. With this knowledge at hand, figure 1-3 shows that a long integer is a 64-bit operand. For an instruction that uses this data type, this is the second operand size since it is not the default operand size of 32 bits and thus would need to be overridden as specified above.

20. In regard to claim 13, Turley discloses the processor as recited in claim 8 wherein the overriding the default operand size with the second operand size is performed in an absence of an operand size override encoding for the first instruction. As shown above, the D bit of the stack segment register specifies the operand size to use for implicit stack operations. This value is not an instruction encoding but actually an overriding default size for implicit operand purposes.

21. In regard to claim 14, Turley discloses the method as recited in claim 8 further comprising overriding the default operand size for the near branch instruction with the second operand size. Page 54 shows that CALL instructions are implicit stack operations. It is inherent in the x86 architecture that CALL instructions are comprised of near and far ("normal" and far, page 180) CALLs and that calls branch to a different piece of code, where near calls branch to code within the current segment and far calls branch to other segments ("near" and "far" pointers, page 15). Because all CALLs are implicit and CALLs include branches, the stack segment operand size overrides the code segment default operand size for near branches.

22. In regard to claim 15, Turley discloses a processor comprising:

a. A circuit configured to generate an indication of a default operand size; Page 45 is the beginning of a chapter on memory segmentation of an Intel 80386 processor circuit. Within this chapter on page 52 it is shown that an indication of default operand size is given in a segment descriptor by a "D bit".

b. And an execution core coupled to receive a near branch instruction, wherein the execution core is configured to override the default operand size with a second operand size responsive to the near branch instruction. It is inherent that the Intel 80386 architecture has an execution core to execute the instruction set that page 10 defines. As shown above, page 52 shows that a D bit gives a default operand size. This section, which starts on page 48, shows that this D bit is in each segment descriptor and therefore specifies the default operand size for each segment descriptor. On pages 266-267, the section entitled "Stack Sizing,"

shows that in the case of an implicit stack operation, the D bit of a *stack* segment descriptor, specifies the address size (and default operand size as shown immediately above for each segment descriptor) instead of the D bit of a *code* segment descriptor as is done for all other instructions. The examiner is taking an instruction with such an implicit stack pointer reference to be the first instruction. This means that the D bit of the stack segment descriptor overrides the default operand size given by the code segment descriptor responsive to the first instruction having implicit stack pointer references. Page 54 shows that CALL instructions are implicit stack operations. It is inherent in the x86 architecture that CALL instructions are comprised of near and far ("normal" and far, page 180) CALLs and that calls branch to a different piece of code, where near calls branch to code within the current segment and far calls branch to other segments ("near" and "far" pointers, page 15). Because all CALLs are implicit and CALLs include branches, the stack segment overrides the code segment default operand size responsive to the instruction being a near branch.

23. In regard to claim 16, Turley discloses the processor as recited in claim 15 further comprising a segment register and a configuration register coupled to the circuit, wherein the segment register is configured to store a segment selector locating a segment descriptor which includes a first operating mode indication and a second operating mode indication, and wherein the configuration register is configured to store an indication, and wherein the circuit is configured to generate the indication of the default operand size responsive to the first operating mode indication, the second

operating mode indication, and the indication in the configuration register. Page 63 in the last paragraph shows that a segment register holds a selector that identifies a segment descriptor in a descriptor table. As shown above, page 52 shows that the segment descriptor has a D bit, which indicates the default operand size to use (16 or 32 bit), and thus specifies (and is responsive to) first and second operating modes specifying whether to operate in 16 or 32 bit default operand size mode. Page 26 shows that there is a Control (configuration) Register 0. It is further shown that this register contains a PE bit that specifies Real or Protected Mode for processor operation. Pages 8-9 show that segmentation is used in Protected Mode and that the reference almost exclusively uses Protected Mode, which covers the segmentation and overriding principles already discussed, until chapter 9. Page 407 in the last paragraph shows that Real Mode was not designed to use these segmentations features of Protected Mode. Therefore, since the Protected Mode is required for the above functionality to work, the indication in the stated control (configuration) register that specifies this mode controls the default operand size, which then is responsive to the configuration register. Also, the first paragraph of page 408 shows that one needs to override the default 16-bit operations (and operands) to use 32 bits ones. Therefore the default operand size is also responsive to the configuration register because the default for Real mode is always 16 bit operands.

24. In regard to claim 17, Turley discloses the processor as recited in claim 16 wherein the default operand size is 32 bits. Turley has shown on page 52 that the default operand size is 32 bits when the D bit is 1.

25. In regard to claim 19, Turley discloses the processor as recited in claim 17 wherein the second operand size is 64 bits. Page 15 shows that figure 1-3 on page 16 gives the data types for use in the processor. It goes on to say that three most common types of operands are the top three. This means that though less common, the other data types are also operands. With this knowledge at hand, figure 1-3 shows that a long integer is a 64-bit operand. For an instruction that uses this data type, this is the second operand size since it is not the default operand size of 32 bits and thus would need to be overridden as specified above.

26. In regard to claim 20, Turley discloses the processor as recited in claim 15 wherein the execution core is configured to override the default operand size with the second operand size in an absence of an operand size override encoding for the near branch instruction. As shown above, the D bit of the stack segment register specifies the operand size to use for implicit stack operations. This value is not an instruction encoding but actually an overriding default size for implicit operand purposes.

27. In regard to claim 21, Turley discloses a method comprising:

a. Generating a default operand size for instructions; Page 45 is the beginning of a chapter on memory segmentation of an Intel 80386 processor circuit. Within this chapter on page 52 it is shown that an indication of default operand size for instructions is given in a segment descriptor by a "D bit".

b. Overriding the default operand size with a second operand size for a near branch instruction. It is inherent that the Intel 80386 architecture has an execution core to execute the instruction set that page 10 defines. As shown

above, page 52 shows that a D bit gives a default operand size. This section, which starts on page 48, shows that this D bit is in each segment descriptor and therefore specifies the default operand size for each segment descriptor. On pages 266-267, the section entitled "Stack Sizing," shows that in the case of an implicit stack operation, the D bit of a *stack* segment descriptor, specifies the address size (and default operand size as shown immediately above for each segment descriptor) instead of the D bit of a *code* segment descriptor as is done for all other instructions. The examiner is taking an instruction with such an implicit stack pointer reference to be the first instruction. This means that the D bit of the stack segment descriptor overrides the default operand size given by the code segment descriptor responsive to the first instruction having implicit stack pointer references. Page 54 shows that CALL instructions are implicit stack operations. It is inherent in the x86 architecture that CALL instructions are comprised of near and far ("normal" and far, page 180) CALLs and that calls branch to a different piece of code, where near calls branch to code within the current segment and far calls branch to other segments ("near" and "far" pointers, page 15). Because all CALLs are implicit and CALLs include branches, the stack segment overrides the code segment default operand size responsive to the instruction being a near branch.

28. In regard to claim 22, Turley discloses the method as recited in claim 21 further comprising generating an operating mode responsive to a first operating mode indication in a segment descriptor and a second operating mode indication in the

segment descriptor and further responsive to an indication in a configuration register, wherein the generating the default operand size is responsive to generating the operating mode. Page 63 in the last paragraph shows that a segment register holds a selector that identifies a segment descriptor in a descriptor table. As shown above, page 52 shows that the segment descriptor has a D bit, which indicates the default operand size to use (16 or 32 bit), and thus specifies (and is responsive to) first and second operating modes specifying whether to operate in (generate) 16 or 32 bit default operand size mode. Therefore, the default operand size is generated responsive to the generation of the operating mode, which specifies the default operand size. Page 26 shows that there is a Control (configuration) Register 0. It is further shown that this register contains a PE bit that specifies Real or Protected Mode for processor operation. Pages 8-9 show that segmentation is used in Protected Mode and that the reference almost exclusively uses Protected Mode, which covers the segmentation and overriding principles already discussed, until chapter 9. Page 407 in the last paragraph shows that Real Mode was not designed to use these segmentations features of Protected Mode. Therefore, since the Protected Mode is required for the above functionality to work, the indication in the stated control (configuration) register that specifies this mode controls the default operand size, which then is responsive to the configuration register. Also, the first paragraph of page 408 shows that one needs to override the default 16-bit operations (and operands) to use 32 bits ones. Therefore the default operand size is also responsive to the configuration register because the default for Real mode is always 16 bit operands.

29. In regard to claim 23, Turley discloses the method as recited in claim 22 wherein the default operand size is 32 bits. Turley has shown on page 52 that the default operand size is 32 bits when the D bit is 1.

30. In regard to claim 25, Turley discloses the method as recited in claim 23 wherein the second operand size is 64 bits. Page 15 shows that figure 1-3 on page 16 gives the data types for use in the processor. It goes on to say that three most common types of operands are the top three. This means that though less common, the other data types are also operands. With this knowledge at hand, figure 1-3 shows that a long integer is a 64-bit operand. For an instruction that uses this data type, this is the second operand size since it is not the default operand size of 32 bits and thus would need to be overridden as specified above.

31. In regard to claim 26, Turley discloses the method as recited in claim 2 wherein the overriding the default operand size with the second operand size is performed in an absence of an operand size override encoding for the near branch instruction. As shown above, the D bit of the stack segment register specifies the operand size to use for implicit stack operations. This value is not an instruction encoding but actually an overriding default size for implicit operand purposes.

Claim Rejections - 35 USC § 103

32. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

33. Claims 4, 11, 18, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Turley in view of Blomgren (5,826,074).

34. In regard to claim 4,

a. Turley discloses the processor as recited in claim 3 wherein the circuit is further configured to generate an indication of a default address size, responsive to the first operating mode indication, the second operating mode indication, and the indication in the configuration register; Page 266 shows that there is an indication of a default address size. Page 264 shows that an address override inverts the sense of the D bit. This means that the D bit not only defines a default operand size, but a default address size as well. As shown above, the D bit gives indications of a first and second operating mode (16 and 32 bit). This means that the default address size is responsive to the first and second operating mode indications. Also as shown above, a control register defines an indication of Real or Protected addressing modes. Page 259 shows that the section described above is for Protected Mode. Page 409 shows that a default is always 16-bit addressing since the override is needed to use 32-bit addressing. Therefore, since the Protected mode can have a default of 16 or 32 bit addressing and the Real mode only 16-bit addressing, the default address size is responsive to the configuration register that specifies these modes.

b. Turley does not disclose wherein the default address size is greater than 32 bits.

c. Blomgren has disclosed a processor where the default address size is greater than 32 bits. Column 3, lines 6-7 show that the processor uses a default 64-bit address size.

d. Blomgren has shown in column 1, lines 45-48 that 64-bit address greatly expands the address space allowing for extended functionality. This greater address size and extended functionality would have motivated one of ordinary skill in the art to change the design of Turley to include 64-bit addressing as taught by Blomgren.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Turley to include the 64-bit addressing scheme taught by Blomgren so that the address space may be greatly expanded to provide for additional functionality.

35. In regard to claim 11,

a. Turley discloses the method as recited in claim 10 further comprising generating a default address size responsive to the generating the operating mode; Page 266 shows that there is a default address size (that is generated). Page 264 shows that an address override inverts the sense of the D bit. This means that the D bit not only defines a default operand size, but a default address size as well. As shown above, the D bit gives indications of a first and second operating mode (16 and 32 bit). This means that the default address size is responsive to the first and second operating mode indications. Also as shown above, a control register defines an indication of the addressing mode. Page 259

shows that the section described above is for Protected Mode. Page 409 shows that a default is always 16-bit addressing since the override is needed to use 32-bit addressing. Therefore, since the Protected mode can have a default of 16 or 32 bit addressing and the Real mode only 16-bit addressing, the default address size is responsive to the configuration register that specifies these modes.

b. Turley does not disclose wherein the default address size is greater than 32 bits.

c. Blomgren has disclosed a processor where the default address size is greater than 32 bits. Column 3, lines 6-7 show that the processor uses a default 64-bit address size.

d. Blomgren has shown in column 1, lines 45-48 that 64-bit address greatly expands the address space allowing for extended functionality. This greater address size and extended functionality would have motivated one of ordinary skill in the art to change the design of Turley to include 64-bit addressing as taught by Blomgren.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Turley to include the 64-bit addressing scheme taught by Blomgren so that the address space may be greatly expanded to provide for additional functionality.

36. In regard to claim 18,

a. Turley discloses the processor as recited in claim 17 wherein the circuit is further configured to generate an indication of a default address size responsive

to the first operating mode indication, the second operating mode indication, and the indication in the configuration register; Page 266 shows that there is an indication of a default address size. Page 264 shows that an address override inverts the sense of the D bit. This means that the D bit not only defines a default operand size, but a default address size as well. As shown above, the D bit gives indications of a first and second operating mode (16 and 32 bit). This means that the default address size is responsive to the first and second operating mode indications. Also as shown above, a control register defines an indication of the addressing mode. Page 259 shows that the section described above is for Protected Mode. Page 409 shows that a default is always 16-bit addressing since the override is needed to use 32-bit addressing. Therefore, since the Protected mode can have a default of 16 or 32 bit addressing and the Real mode only 16-bit addressing, the default address size is responsive to the configuration register that specifies these modes.

b. Turley does not disclose wherein the default address size is greater than 32 bits.

c. Blomgren has disclosed a processor where the default address size is greater than 32 bits. Column 3, lines 6-7 show that the processor uses a default 64-bit address size.

d. Blomgren has shown in column 1, lines 45-48 that 64-bit address greatly expands the address space allowing for extended functionality. This greater address size and extended functionality would have motivated one of ordinary

skill in the art to change the design of Turley to include 64-bit addressing as taught by Blomgren.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Turley to include the 64-bit addressing scheme taught by Blomgren so that the address space may be greatly expanded to provide for additional functionality.

37. In regard to claim 24,

a. Turley discloses the method as recited in claim 23 further comprising generating a default address size responsive to the generating the operating mode; Page 266 shows that there is a default address size (that is generated). Page 264 shows that an address override inverts the sense of the D bit. This means that the D bit not only defines a default operand size, but a default address size as well. As shown above, the D bit gives indications of a first and second operating mode (16 and 32 bit). This means that the default address size is responsive to the first and second operating mode indications. Also as shown above, a control register defines an indication of the addressing mode. Page 259 shows that the section described above is for Protected Mode. Page 409 shows that a default is always 16-bit addressing since the override is needed to use 32-bit addressing. Therefore, since the Protected mode can have a default of 16 or 32 bit addressing and the Real mode only 16-bit addressing, the default address size is responsive to the configuration register that specifies these modes.

b. Turley does not disclose wherein the default address size is greater than 32 bits.

c. Blomgren has disclosed a processor where the default address size is greater than 32 bits. Column 3, lines 6-7 show that the processor uses a default 64-bit address size.

d. Blomgren has shown in column 1, lines 45-48 that 64-bit address greatly expands the address space allowing for extended functionality. This greater address size and extended functionality would have motivated one of ordinary skill in the art to change the design of Turley to include 64-bit addressing as taught by Blomgren.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Turley to include the 64-bit addressing scheme taught by Blomgren so that the address space may be greatly expanded to provide for additional functionality.

Conclusion

38. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

39. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following patents and publication have been cited to further show the art with respect to operand size overriding.

US Pat No 4,926,322 to Stimac shows that operand size, address size, and segment overriding mechanisms and methods.

US Pat No 5,644,755 to Wooten discloses segment and operand override instruction encodings controlled by a segment descriptor.

US Pat No 5,758,116 to Lee teaches operand override prefixes that override a default operand size.

US Pat No 6,105,125 to Nemirovsky describes a method and apparatus for overriding default operand sizes for implicit stack operations and near branches in his background

Intel's Embedded 486 Processor Family Developer's Manual discloses extensive operand override functionality.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
February 10, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100